CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Kyoto University Research Information Repository
KURENAI
Kyoto University Research Information Repository
KYOTO UNIVERSITY

Title	Analysis of an intrusion tolerant database system via semi-Markov processes (Theory and Application of Decision Analysis in Uncertain Situation)
Author(s)	Uemura, Toshikazu; Dohi, Tadashi
Citation	数理解析研究所講究録 (2008), 1589: 241-250
Issue Date	2008-04
URL	http://hdl.handle.net/2433/81564
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

Kyoto University

数理解析研究所講究録
第1589巻 2008年 241-250

Analysis of an intrusion tolerant database system via semi-Markov processes

植村俊和, 土肥正
Toshikazu Uemura and Tadashi Dohi
Department of Information Engineering, Graduate School of Engineering,
Hiroshima University, Japan

1. Introduction

The use of computer-based systems and Internet has been undergoing dramatic growth in scale, variety and penetration, implying our growing dependence on them for a large number of businesses and day-to-day life services. Unfortunately, the complexity, the heterogeneity and the openness of the supporting infrastructures to untrusted users have also given rise to an increasing number of vulnerabilities and malicious threats (viruses, worms, denial of service attacks, fishing attempts, *etc.*). For malicious attackers, if the access right strengthens, the probability that the security intrusion may happen will effectively decrease, but the utilization on accessibility will be rather lost. The classical security-related work has traditionally privileged, with a few exceptions, *intrusion avoidance techniques* (vulnerability elimination, strong authentication, *etc.*) and *attack deterrence* (attack tracing, auditing, *etc.*). However, such techniques have proved to be not sufficient to ensure the security of systems connected to networks.

More recently, *intrusion tolerance techniques*, inspired from traditional techniques commonly used for tolerating accidental faults in hardware and/or software systems, have received considerable attention to complement intrusion avoidance techniques, and improve the security of systems connected to the Internet. So far, most efforts in security have been focused on specification, design and implementation issues. In fact several implementation techniques of intrusion tolerance at the architecture level have been developed for real computer-based systems such as distributed systems [1], database systems [6,7], middleware [15,16], server systems [2]. The above implementation approaches are based on the redundant design at the architecture level on secure software systems. In other words, since these methods can be categorized by a design diversity technique in secure system design and need much cost for the development, the effect on implementation has to be evaluated carefully and quantitatively.

The quantitative evaluation of information security based on modeling is becoming much popular to validate the effectiveness of computer-based systems with intrusion tolerance. Littlewood *et al.* [5] found the analogy between the information security theory and the traditional reliability theory in assessing the quantitative security of operational software systems, and explored the feasibility of probabilistic quantification on security. Jonsson and Olovsson [4] gave a quantitative method to study the attacker's behavior with the empirical data observed in experiments. Ortalo, Deswarte and Kaaniche [11] applied the privilege graph and the continuous-time Markov chain (CTMC) to evaluate the system vulnerability, and derived the mean effort to security failure. Singh, Cukier and Sanders [12] designed stochastic activity networks model for probabilistic validation of security and performance of several intrusion tolerant architectures. Stevens *et al.* [13] also proposed probabilistic methods to model the DPASA (Designing Protection and Adaptation into a Survivable Architecture).

On the other hand, it would be quite effective to apply the traditional Markov/semi-Markov modeling

approaches to design the state transition diagram of system security states by incorporating both attacker and system behaviors under uncertainty. Madan *et al.* [9] dealt with an architecture with intrusion tolerance, called SITAR (Scalable Intrusion Tolerant Architecture) and described the stochastic behavior of the system by discrete-time semi-Markov chain (DTSMC). They also derived analytically the mean time length to security failure. Imaizumi, Kimura and Yasui [3] and Uemura and Dohi [14] focused on the typical denial of service attacks for server systems and formulated the optimization problems on the optimal monitoring time and the optimal patch management policy via continuous-time semi-Markov chain (CTSMC) models. Although they mainly considered the expected cost models which are familiar to the Markov/semi-Markov analyses, the relationship with security attributes was still unclear in modeling.

For the purpose of comprehensive modeling of system-level security quantification, it is actually difficult to model certain security attributes such as *confidentiality* and *integrity* using the probabilistic techniques as well as to quantify the high-level security requirement with different security attributes [10]. Hence, the measurement techniques for model parameterization and validation must be carefully selected in security evaluation. In such a situation, the *survivability* analysis is becoming very common to quantify the computer-based systems under the assumption that failure may occur and that the outcome of the failure negatively impacts a large segment of the subscribers to the IT infrastructure, where such failures may be the result of deliberate, malicious attacks against the infrastructure by an adversary.

In this paper we consider the secure design of an intrusion tolerant database (ITDB) system with a control parameter, and describe the stochastic behavior of an intrusion tolerant database system (ITDB). First, Liu *et al.* [6,7] proposed several ITDB architectures and presented the design and implementation methodologies. While traditional secure database systems rely on preventive controls and are very limited in surviving malicious attacks, the ITDB can detect intrusions and isolate attacks. In addition, it can contain, assess and repair the damage caused by intrusions in a timely manner such that sustained, self-stabilized levels of data integrity and availability can be provided to applications in the face of attacks. With the aim to quantify the ITDB, Yu, Liu and Zang [18] and Wang and Liu [17] developed simple CTMC models to evaluate the survivability of the ITDB. Especially, Wang and Liu [17] formulated two survivability measures; system integrity and rewarding availability[1]. In this paper we extend it to a CTSMC model with non-exponentially distributed transition times, and provide more robust quantitative framework to malicious attacks with a variety of probabilistic patterns.

Further, by introducing an additional control parameter called the *switching time*, we develop secure control schemes of the ITDB, which maximize the security measures; system integrity and rewarding availability, as well as the common system availability. Necessary and sufficient conditions to exist a finite and unique optimal switching time are derived under a mild parametric assumption. These analytical results enable us to maximize the utility of intrusion tolerance in the ITDB. Numerical examples are devoted to examine the dependence of model parameters on the optimal switching time and its associated security measures. Throughout the sensitivity analysis on the model parameters, it is shown numerically that the ITDB should be designed to minimize mission impact by containing both the intrusion and failure. Finally, the paper is concluded with some remarks and future research directions.

[1]The *integrity* defined in [17] seems to be somewhat different from the usual qualitative definition as a security attribute. In this paper we call it the *system integrity* which is a quantitative measure, and distinguish from the qualitative measure.

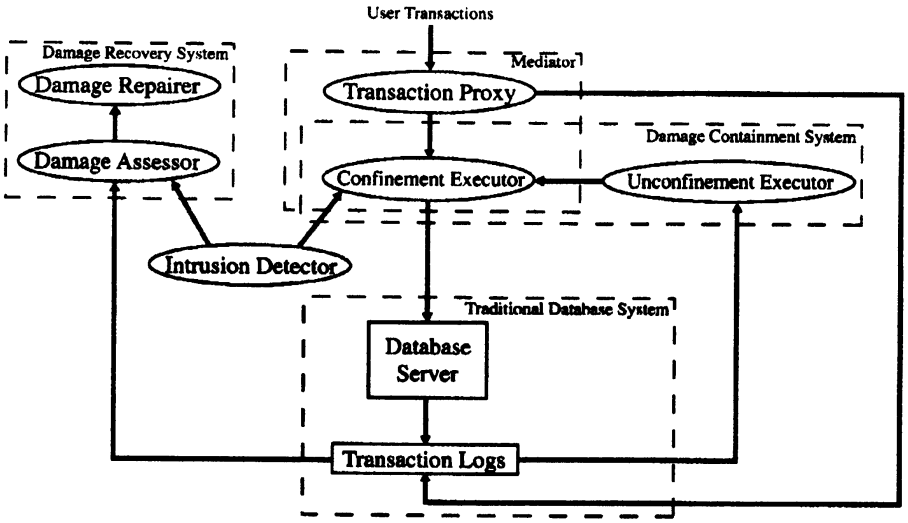

Figure 1: Basic ITDB architecture.

2. Intrusion Tolerant Database System

2.1. Basic Concept

First of all, we give a brief summary on the intrusion tolerant database (ITDB). In the ITDB, once it is damaged from any reason such as infections and attacks, the damaged parts are automatically located, contained and repaired as soon as possible, so that the database can continue being operative with the intrusion tolerant functions. Figure 1 shows the major components of a comprehensive ITDB, which was introduced in [6, 7]. In a fashion similar to the reference [17], we also focus on some significant components; *Mediator*, *Damage containment* and *Damage recovery*, in Fig.1 and describe the stochastic behavior of functions in major components. Mediator subsystem may function as a *proxy* for each user transaction and transaction processing call to the database system, and enables to keep the useful information on user transactions, such as read/write operations. This function is quite important to generate the corresponding logs for damage recovery and containment.

More precisely, in the traditional secure database system, the damage containment can not be made until the data items are identified as damaged ones. In this situation, a significant damage assessment latency may happen, so that the damage caused by attacks or intrusions may propagate to the other data items. In the ITDB, the so-called *multi-phase damage containment* technique is applied as an intrusion tolerant technique [6], where it involves one containing phase and one more uncontaining phases referred to as *Containment relaxation*. Once an intrusion is detected by *Intrusion detector*, Damage recovery subsystem has the responsibility to the damage assessment and repair, and retrieves the malicious transaction messages reported from Intrusion detector. On the other hand, Damage containment subsystem traces the damage propagation by capturing the dependent-upon relationship among transactions.

Hence, the control by Intrusion detector plays an central role to the design of the ITDB. Since Intrusion detector is based on both the trails on the logs and some relevant rules to identify malicious transactions, however, its effect is limited. In other words, it would be impossible to detect all the intrusions automatically within the real time. In practice, two control modes can be ready; automatic detection mode and manual detection mode, so that an automatic detection mode can be switched to

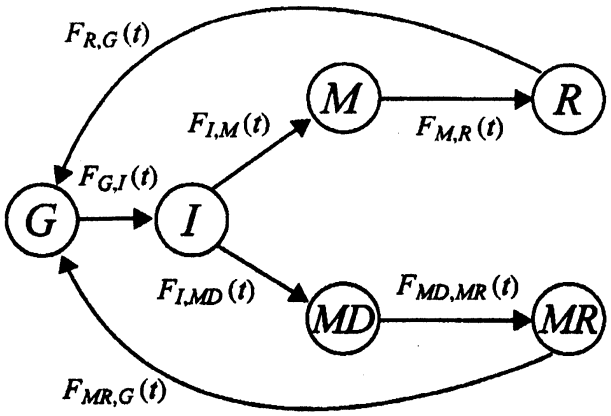

Figure 2: Semi-Markov transition diagram

a manual detection mode if Intrusion detector does return no response during the real time operation. Wang and Liu [17] developed a simple CTMC model with random switching from an automatic detection mode to a manual one, and evaluated the security measures for the ITDB.

2.2. Model Description

Following Wang and Liu [17], we also focus on three components in the ITDB, Mediator, Damage recovery and Damage containment systems. Suppose that the database system starts operating at time $t = 0$ with *Normal State*; G. If attackers or hackers detect the vulnerability of the database, they try to attack the database and the state may make a transition to *Infection State*; I, where the transition time from G to I has the continuous cumulative distribution function (c.d.f.) $F_{G,I}(t)$ with mean $\mu_{G,I}$ (> 0). Once the malicious attack by an attacker was successful in State I, the intrusion detector begins operating automatically. If the infection of parts or data items is detected in the automatic detection mode, the state makes a transition from I to *Maintenance State*; M, where the transition time from I to M is given by a random variable having the continuous c.d.f. $F_{I,M}(t)$ and mean $\mu_{I,M}$ (> 0). In this phase, when the infected items are identified more specifically through the damage assessor, the corrective recovery operation is triggered in *Recovery State*; R in the damage recovery system. Let the state transition time from M to R be the random variable having the c.d.f. $F_{M.R}(t)$ and mean $\mu_{M,R}$ (> 0). After the completion of recovery operation, the infected parts are fixed and the database system can become as good as new with Normal State, where the completion time to recover the database is given by the non-negative continuous random variable with the c.d.f. $F_{R,G}(t)$ and mean $\mu_{R,G}$ (> 0).

On the other hand, it should be worth mentioning that the infection of parts or data items is not always possible only in the automatic detection mode. In other words, the intrusion detection is not always perfect for all possible attacks, so that the system manager and/or the full vendor may search the infected parts in the manual detection mode. Wang and Liu [17] considered the possibility of switching from the automatic detection mode to the manual detection mode, and assumed that the switching may occur randomly. This corresponds to the switching from the unconfinement executor to the confinement executor. In [17], the associated stochastic model is based on a CTMC with exponentially distributed transition times. Instead of the exponential switching time, we model the switching time by the non-negative continuous random variable with the c.d.f. $F_{I,MD}(t)$ and mean $\mu_{I,MD}$ (> 0), where *Manual detection state* is denoted by MD, and the damaged parts are contained manually within the ITDB.

When the intrusion is detected, the system state makes transition from MD to MR, and next the recovery operation starts immediately. Finally, when the recovery operation is complete, the state makes a transition from MR to G with Normal State. In this way, the same cycle repeats again and again over an infinite time horizon. Since the underlying stochastic process is a CTSMC, it is noted that our model is an extended version to the CTMC model in [17]. Figure 2 illustrates the state-transition diagram for the CTSMC model.

In this context, the automatic detection mode is randomly switched to the manual detection mode. Dissimilar to Wang and Liu [17], we introduce the time limit to turn on the manual detection, t_0 ($0 \leq t_0 < \infty$), periodically and call it the *switching* time. If the automatic detection is switched to the manual detection, then the system state goes to I from MD. Without any loss of generality, we define the transition probability from I to MD by

$$F_{I,MD}(t) = \begin{cases} 1 & (t \geq t_0) \\ 0 & (t < t_0). \end{cases} \tag{1}$$

This means that the detection mode can be switched from the automatic mode to the manual model at every t_0 time unit.

3. Security Measures

3.1. System Integrity

Wang and Liu [17] defined the system integrity as a fraction of time when all accessible data items in the database are clean. As mentioned previously in Section 1, the integrity is regarded as one of the most typical security attributes in addition to authentication and non-repudiation. When the integrity is high, the ITDS can serve the users by utilizing the good or clean data with high probability. In Fig. 2, all data items in the ITDB are clean and accessible in State G. When attacks occur, some data items will be affected and the part of accessible data items in state I may be *dirty*. After the intrusion is identified, the ITDB can contain all the damaged data until it finishes the repair process. In this situation, the ITDB carries out the selective containment and repair, and is still available, so that the accessible data items are clean during the containment, damage assessment and repair process. In Fig. 2, since the system states under consideration are G, M, R and MR, the system integrity is defined by $IN(t_0) = U_{IN}(t_0)/T(t_0)$, where

$$U_{IN}(t_0) = \mu_{G,I} + (\mu_{M,R} + \mu_{R,G})F_{I,M}(t_0) + \mu_{MR,G}\overline{F}_{I,M}(t_0), \tag{2}$$

$$T(t_0) = U_{IN}(t_0) + \int_0^{t_0} \overline{F}_{I,M}(t)dt + \mu_{MD,MR}\overline{F}_{I,M}(t_0). \tag{3}$$

Then, the problem is to derive the optimal switching time t_0^* maximizing $AV(t_0)$. For the purpose, we make the following parametric assumption:

(A-1) $\mu_{MR,G} > \mu_{M,R} + \mu_{R,G}$.

In (A-1), it is assumed that the time length to detect an intrusion automatically is strictly shorter than that by the manual detection. This seems to be intuitively validated from the viewpoint of the utility in automatic detection.

Proposition 1: (1) Suppose that the c.d.f. $F_{I,M}(t)$ is strictly DHR under (A-1). Define the function:

$$q_{IN}(t_0) = (\mu_M + \mu_R - \mu_{MR})r_D(t_0)T_{IN}(t_0)$$

$$-\Big[1+\{(\mu_M+\mu_R)-(\mu_{MD}+\mu_{MR})\}r_D(t_0)\Big]U_{IN}(t_0). \tag{4}$$

(i) If $q_{IN}(0) > 0$ and $q_{IN}(\infty) < 0$, then there exists a finite and unique optimal switching time t_0^* $(0 < t_0^* < \infty)$ satisfying $q_{IN}(t_0^*) = 0$

(ii) If $q_{IN}(0) \leq 0$, then $t_0^* = 0$

(iii) If $q_{IN}(\infty) \geq 0$, then $t_0^* \to \infty$

(2) Suppose that the c.d.f. $F_{I,M}(t)$ is IHR under (A-1). If $IN(0) > IN(\infty)$, then $t_0^* = 0$ otherwise $t_0^* \to \infty$.

The proof is omitted for brevity. For the actual management of database systems, it is more significant to keep the clean and accessible data. So, when the quality of data is considered, the system integrity should be the more attractive security measure than the system availability.

3.2. Rewarding Availability

The system availability is defined as a fraction of time when the ITDB is providing services to its users, and does not care the quality of data. Since the ITDB performs the on-the-fly repair and will not stop its service faced by attacks, it can be expected that the corresponding system availability is nearly 100% in almost all cases. For better evaluation of the security attribute in the ITDB, Wang and Liu [17] considered another type of availability, called *rewarding availability*, which is defined as a fraction of time when all the clean data items are accessible. If the clean data can not be accessed in the ITDB, it can be regarded as a serious loss of service to users. Dissimilar to the system integrity, since the system states under consideration are G, R and MR, the rewarding availability is defined by $RA(t_0) = U_{RA}(t_0)/T(t_0)$, where

$$U_{RA}(t_0) = \mu_{G,I} + \mu_{R,G}F_{I,M}(t_0) + \mu_{MR,G}\overline{F}_{I,M}(t_0). \tag{5}$$

We give the characterization result on the optimal switching time maximizing the rewarding availability without the proof.

Proposition 2: (1) Suppose that the c.d.f. $F_{I,M}(t)$ is strictly DHR under (A-1). Define the function:

$$\begin{aligned} q_{RA}(t_0) &= (\mu_{R,G}-\mu_{MR,G})r_{I,M}(t_0)T(t_0) \\ &\quad -\Big[1+\{(\mu_{M,R}+\mu_{R,G})-(\mu_{MD,MR}+\mu_{MR,G})\}r_{I,M}(t_0)\Big]U_{RA}(t_0). \end{aligned} \tag{6}$$

(i) If $q_{RA}(0) > 0$ and $q_{RA}(\infty) < 0$, then there exists a finite and unique optimal switching time t_0^* $(0 < t_0^* < \infty)$ satisfying $q_{RA}(t_0^*) = 0$

(ii) If $q_{RA}(0) \leq 0$, then $t_0^* = 0$

(iii) If $q_{RA}(\infty) \geq 0$, then $t_0^* \to \infty$

(2) Suppose that the c.d.f. $F_{I,M}(t)$ is IHR under (A-1). If $RA(0) > RA(\infty)$, then $t_0^* = 0$ otherwise $t_0^* \to \infty$.

In this section, we optimized the three security measures for the ITDB and derived the optimal switching times for respective quantitative criteria. In the following section, we will give some numerical examples, and calculate the optimal switching policies and their associated security measures.

Table 1: Model parameters.

Parameters	Values
attack hitting rate (λ_a)	0.5 (low); 1 (moderate); 5 (heavy)
detection rate ($\lambda_{I,M}$)	10 (slow); 15 (medium); 20 (fast)
marking rate ($\lambda_{M,R}$)	27
repair rate ($\lambda_{R,G}$)	22
manual detection rate ($\lambda_{MD,MR}$)	0.02
manual repair rate ($\lambda_{MR,G}$)	0.02
false alarm rate (α)	10%, 20%, 50%

4. Numerical Illustrations

4.1. Parameter Set

We focus on both the system integrity and the rewarding availability, and treat the database management system with Oracle 9*i* server in [17]. Although the security model in [17] was based on a simple CTMC, we here assume that the c.d.f. $F_{I,M}(t)$ is given by the Weibull distribution with scale parameter η and shape parameter m:

$$F_{I,M}(t) = 1 - \exp\{-(t/\eta)^m\}. \tag{7}$$

This assumption implies that the transition time from an intrusion to the containment sate is DHR ($m \leq 1$) or IHR ($m \geq 1$), and can represent the more general transition phenomena. When $m = 1$, it reduces to the exponential distribution with constant hazard rate. The other transition rates from state i to state j are assumed to constant, *i.e.*, $1/\mu_{i,j} = \lambda_{i,j}$ $(i, j \in \{G, I, M, R, MD, MR\},\ i \neq j)$, except for $(i, j) = (I, M)$. In particular, we introduce the attack hitting rate λ_α and the false alarm rate α as Wang and Liu [17] did so. It should be noted that Intrusion detector in Fig. 1 will warn the system user of malicious attacks/intrusions as well as the system failure by means of a false alarm. Let T_a and T_{fa} be the intrusion time and the system failure time measured from time $t = 0$ in State G, and be the exponentially distributed random variables with parameters λ_a and α, respectively. Then the function $F_{G,I}(t)$ is regarded as the c.d.f. of the random variable $\min\{T_a, T_{fa}\}$ and is the exponential c.d.f. with parameter $\lambda_a + \alpha$. Table 1 presents the model parameters used in this example, where they are almost same in [17]. We set $m = 0.2$, and choose η so as to satisfy $\mu_{I,M} = \eta\Gamma(1 + 1/m)$.

4.2. System Integrity

Table 2 presents the maximized the system integrity for varying model parameters, where $t_0 \to \infty$ implies the no-manual detection policy. From this table, it is seen that the optimal control of the switching time to the manual detection mode leads to the 2.8% $\sim$ 35.5% improvement of system integrity. In this numerical example, it can be observed that the periodic switching to the manual detection mode and the rapid containment/repair from the damage due to attacks or intrusions are quite important factors to increase the system integrity. In Fig.3, we plot the behavior of the system integrity with respect to the attack hitting rate and the false alarm rate. From this result, it can be seen that the system integrity increases to 0.2% $\sim$ 1.4% (1.3×10^{-2}% $\sim$ 0.16%) when the attack hitting rate (false alarm rate) decreases. This result can be explained physically, so that the system integrity can increase if the total operation time of the ITDB becomes longer with the lower attack hitting rate and/or if the load of the ITDB with the higher false alarm rate becomes smaller.

Table 2: Maximizing system integrity for varying model parameters.

$(\lambda_{I,M}, \lambda_a, \alpha)$	$t_0 \to \infty$	t_0^*	$IN(t_0^*)$	increment (%)
(10,0.5,10)	0.9459	104.3340	0.9975	5.4505
(10,0.5,50)	0.9154	104.1290	0.9959	8.7926
(10,5.0,10)	0.7358	102.5530	0.9841	33.7328
(10,5.0,50)	0.7255	102.4370	0.9832	35.5193
(15,0.5,10)	0.9633	115.7010	0.9991	3.7195
(15,0.5,50)	0.9420	115.6220	0.9986	6.0068
(15,5.0,10)	0.8069	115.0200	0.9944	23.2411
(15,5.0,50)	0.7986	114.9760	0.9941	24.4872
(20,0.5,10)	0.9722	124.4060	0.9996	2.8182
(20,0.5,50)	0.9559	124.3680	0.9994	4.5527
(20,5.0,10)	0.8478	124.0820	0.9975	17.6578
(20,5.0,50)	0.8409	124.0620	0.9974	18.6079

Table 3: Maximizing rewarding availability for varying model parameters.

$(\lambda_{I,M}, \lambda_a, \alpha)$	$t_0 \to \infty$	t_0^*	$RA(t_0^*)$	increment (%)
(10,0.5,10)	97.8264	0.9506	0.9259	2.6735
(10,0.5,50)	93.5265	0.9213	0.8841	4.2076
(10,5.0,10)	55.5619	0.7158	0.6380	12.1884
(10,5.0,50)	52.1533	0.7008	0.6238	12.3416
(15,0.5,10)	108.6100	0.9529	0.9429	1.0565
(15,0.5,50)	104.1020	0.9249	0.9098	1.6660
(15,5.0,10)	66.1144	0.7347	0.6996	5.0182
(15,5.0,50)	63.0064	0.7219	0.6867	5.1321
(20,0.5,10)	116.8210	0.9535	0.9516	0.1983
(20,0.5,50)	112.0580	0.9260	0.9231	0.3088
(20,5.0,10)	72.5066	0.7404	0.7351	0.7254
(20,5.0,50)	69.3535	0.7282	0.7231	0.7069

4.3. Rewarding Availability

Similar to Subsection 4.2, we examine the dependence of model parameters on the optimal switching time and its associated rewarding availability in Table 3. From this table, it can be found that the periodic control on the switching to the manual detection mode enables us to increase the rewarding availability up to 0.2% $\sim$ 12.3%. As the detection speed becomes faster, it can be increased to 0.3% $\sim$ 3.9%. Figure 4 shows the behavior of rewarding availability on the attack hitting rate and the false alarm, where the rewarding availability varies in the ranges of 27.2% $\sim$ 32.8% and 1.7% $\sim$ 3.2% for α and λ_α, respectively. Thus, the attack hitting rate is more sensitive than the false alarm rate to not only the system integrity but also the rewarding availability.

5. Conclusions

In this paper we have reconsidered an ITDB architecture in Wang and Liu [17] and developed a CTSMC to assess the security measures such as system availability, system integrity and rewarding availability. Further, we have optimized the switching times for maximizing the above measures and given the optimal design methodologies in terms of intrusion tolerance. In numerical examples, we have calculated the optimal switching times and their associated security measures, and carried out the sensitivity analysis on model parameters. As the lesson learned from the numerical examples, it has been shown that the system integrity and the rewarding availability could be improved by controlling appropriately the switching times to the manual detection mode.

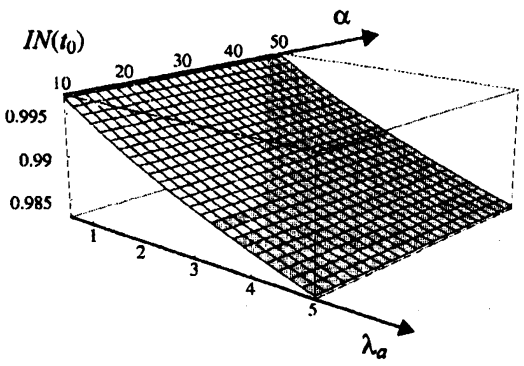

Figure 3: Behavior of system integrity with respect to λ_a and α.

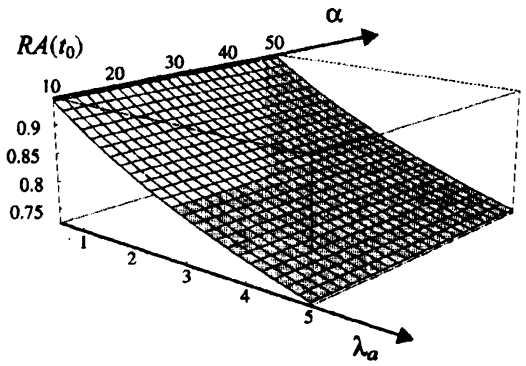

Figure 4: Behavior of rewarding availability with respect to λ_a and α.

In the on-going research, we will evaluate quantitatively the other measures in survivability in the ITDB. Since the survivability can be evaluated in the same framework as performability [?, 10], the CTSMC model developed in this paper can be still useful for the analysis with different measures. Also, though we focused on only Mediator subsystem as a proxy for each user transaction and transaction processing call to the database system, the other part on dynamic transaction processing such as the database system itself may be included for modeling from the macroscopic point of view. Such an integrated model should be developed by applying the semi-Markov analysis in the future.

References

[1] Y. Deswarte, L. Blain and J. C. Fabre, "Intrusion tolerance in distributed computing systems," *Proceedings of 1991 IEEE Symposium on Research in Security and Privacy*, pp. 110–121, IEEE CS Press (1991).

[2] V. Guputa, V. Lam, H. V. Ramasamy, W. H. Sanders and S. Singh, "Dependability and performance evaluation of intrusion-tolerant server architectures," *LADC 2003*, LNCS **2847**, pp. 81–101, Springer-Verlag (2003).

[3] M. Imaizumi, M. Kimura and K. Yasui, "Reliability analysis of a network server system with illegal access," *Advanced Reliability Modeling II* (W. Y. Yun and T. Dohi, eds.), pp. 40–47, World Scientific (2006).

[4] E. Jonsson and T. Olovsson, "A quantitative model of the security intrusion process based on attacker behavior," *IEEE Transactions on Software Engineering*, **23** (4), pp. 235–245 (1997).

[5] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Doboson, J. McDermid and D. Gollmann, "Towards operational measures of computer security," *Journal of Computer Security*, **2** (2/3), pp. 211–229 (1993).

[6] P. Liu, "Architectures for intrusion tolerant database systems," *Proceedings of 18th Annual Computer Security Applications Conference* (ACSAC 2002), pp. 311–320, IEEE CS Press (2002).

[7] P. Liu, J. Jing, P. Luenam, Y. Wang, L. Li and S. Ingsriswang, "The design and implementation of a self-healing database system," *Journal of Intelligent Information Systems*, **23** (3), pp. 247–269 (2004).

[8] Y. Liu, V.B. Mendiratta, and K. Trivedi, "Survivability analysis of telephone access network," *Proceedings of 15th International Symposium on Software Reliability Engineering* (ISSRE 2004), pp. 367–377, IEEE CS Press (2004).

[9] B. B. Madan, K. Goseva-Popstojanova, K. Vaidyanathan, and K. S. Trivedi, "A method for modeling and quantifying the security attributes of intrusion tolerant systems,", *Performance Evaluation*, **56** (1/4), pp. 167–186 (2004).

[10] D. M. Nikol, W. H. Sanders and K. S. Trivedi, "Model-based evaluation: from dependability to security," *IEEE Transactions on Dependability and Secure Computing*, **1** (1), pp. 48–65 (2004).

[11] R. Ortalo, Y. Deswarte and M. Kaaniche, "Experimenting with quantitative evaluation tools for monitoring operational security," *IEEE Transactions on Software Engineering*, **25** (5), pp. 633–650 (1999).

[12] S. Singh, M. Cukier and W. H. Sanders, "Probabilistic validation of an intrusion tolerant replication system," *Proceedings of 33rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (DSN 2003), pp. 615–624, IEEE CS Press (2003).

[13] F. Stevens, T. Courtney, S. Singh, A. Agbaria, J. F. Meyer, W. H. Sanders and P. Pal, "Model-based validation of an intrusion-tolerant information system," *Proceedings of 23rd IEEE Reliable Distributed Systems Symposium* (SRDS 2004), pp. 184–194, IEEE CS Press (2004).

[14] T. Uemura and T. Dohi, "Quantitative evaluation of intrusion tolerant systems subject to DoS attacks via semi-Markov cost models," *Emerging Directions in Embedded and Ubiquitous Computing: International Conference EUC 2007 Workshops* (M. K. Denko, C.-S. Shih, K.-C. Li, S.-L. Tsao, Q.-A. Zeng, S.-H. Park, Y.-B. Ko, S.-H. Hung and J.-H. Park, eds.), LNCS **4809**, pp. 31–42, Springer-Verlag (2007).

[15] P. E. Verissimo, N. F. Neves and M. Correia, "Intrusion-tolerant architectures: concepts and design," *Architecting Dependable Systems* (R. Lemos, C. Gacek and A. Romanovsky, eds.), LNCS **2677**, pp. 3–36, Springer-Verlag (2003).

[16] P. E. Verissimo, N. F. Neves, C. Cachin, J. Poritz, D. Powell, Y. Deswarte, R. Stroud and I. Welch, "Intrusion-tolerant middleware," *IEEE Security and Privacy*, **4** (4), pp. 54–62 (2006).

[17] H. Wang and P. Liu, "Modeling and evaluating the survivability of an intrusion tolerant database system," *ESORICS 2006* (D. Gollmann, J. Meier and A. Sabelfeld, eds.), LNCS **4189**, pp. 207–224, Springer-Verlag (2006).

[18] M. Yu, P. Liu and W. Zang, "Self-healing workflow systems under attacks," *Proceedings of 24th International Conference on Distributed Computing Systems* (ICDCS 2004), pp. 418–425, IEEE CS Press (2004).